

**NASA Technical Memorandum 104137**

11-2  
-7375  
P-8

**UNSTRUCTURED-GRID METHODS DEVELOPMENT -  
LESSONS LEARNED**

(NASA-TM-104137) UNSTRUCTURED-GRID METHODS  
DEVELOPMENT: LESSONS LEARNED (NASA) 8 p  
CSCL 01A

N91-50000

Unclass

65/02 0037375

**JOHN T. BATINA**

**AUGUST 1991**



National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23665



## UNSTRUCTURED-GRID METHODS DEVELOPMENT - LESSONS LEARNED

John T. Batina  
NASA Langley Research Center  
Hampton, Virginia 23665-5225 USA

### ABSTRACT

The paper summarizes the development by the author of unstructured grid methods for the solution of the equations of fluid flow and shares some of the "lessons learned" over the course of the research. The focus of the discussion is on the solution of the three-dimensional Euler equations including spatial discretizations, temporal discretizations, and boundary conditions. An example calculation with an upwind implicit method using a CFL number of infinity is presented for the Boeing 747 aircraft. The results were obtained in less than one hour CPU time on a Cray-2 computer, thus demonstrating the speed and robustness of the present capability.

### INTRODUCTION

The development of unstructured-grid methods for the solution of the equations of fluid flow has become quite popular in recent years because of the advantages that these methods offer in comparison with the more-traditional structured-grid methods [1-19]. For example, with an unstructured grid, it is relatively easy to model very complicated three-dimensional geometries such as a complete aircraft configuration. With a structured grid, it is generally much more difficult to achieve this level of geometrical complexity without resorting to more sophisticated meshing methodologies (such as blocked, patched, chimera, or hybrid type grids), which in turn, significantly complicate the solution algorithm of the governing fluid flow equations. A second advantage, is that unstructured grid methods enable in a natural way for adaptive mesh refinement to more accurately predict the physics of the flow. These methods involve enrichment and coarsening procedures to either add points in high gradient regions of the flow or remove points where they are not needed, to produce solutions of high spatial accuracy at minimal computational cost. Furthermore, analogous to spatial adaption, a temporal adaption technique may be used on an unstructured grid to improve the computational efficiency of explicit time-marching schemes for unsteady aerodynamic applications. Temporal adaption can be thought of as time-accurate local time stepping. With this procedure, each grid cell is integrated according to the local flow physics and numerical stability. Time accuracy is maintained by bringing all cells to the same time level as determined by the step size of the largest cell.

The author has been involved over the past four years in the development of unstructured-grid methods for the solution of the time-dependent Euler and Navier-Stokes equations [10-19]. These methods are being developed primarily for unsteady aerodynamic and aeroelastic applications. As a result of developing and testing a variety of methods, much knowledge and experience has been gained. Therefore the purpose of the paper is to summarize this developmental effort and to share some of the "lessons learned" over the course of the research. It is heretofore emphasized that the views presented are those of the author, which are not necessarily consistent with the views of other researchers. Differences in opinion over the best choice of method, for example, may result due to the unsteady application of the present methods. The focus of the discussion is on the solution of the three-dimensional Euler equations including spatial discretizations, temporal discretizations, and boundary conditions. Both central-difference and upwind-type spatial discretizations are described in the context of both cell-vertex and cell-centered implementations. Explicit and implicit temporal discretizations are given along with methods to accelerate convergence to steady state. Strong and weak implementations of the surface boundary conditions and far field conditions are also described. Finally, an example calculation is presented for the Boeing 747 aircraft to demonstrate the speed and robustness of the present capability.

### EULER EQUATIONS

In the present study the flow is assumed to be governed by the three-dimensional time-dependent Euler equations which may be written in integral form as

$$\frac{\partial}{\partial t} \int_{\Omega} Q dV + \int_{\partial\Omega} (En_x + Fn_y + Gn_z) dS = 0 \quad (1)$$

where  $Q$  is the vector of conserved variables, and  $E$ ,  $F$ , and  $G$  are the convective fluxes. Equation (1) has been nondimensionalized by the freestream density and the freestream speed of sound. Also, the second integral in Eq. (1) is a boundary integral resulting from application of the divergence theorem, and  $n_x$ ,  $n_y$ , and  $n_z$  are Cartesian components of the unit normal to the boundary surface.

## SPATIAL DISCRETIZATIONS

In general, the Euler equations in integral form have been solved numerically using several different finite-volume spatial discretizations on unstructured grids of tetrahedral cells [12, 16, 17]. These discretizations are either of the central-difference type with explicit artificial dissipation or of the upwind-type which are naturally dissipative.

### Central-Difference Algorithms

Central-difference type algorithms may be implemented as either cell-vertex (node-based) or cell-centered discretizations. In the cell-vertex approach, the flow variables are stored at the nodes and the control volume at a given node is typically taken to be all of the neighboring cells which have a vertex at that node. In the cell-centered approach, the flow variables are stored at the centroids of the cells and the control volume is simply the cell itself. It is noted that, in general, tetrahedral meshes contain between five and six times the number of cells as nodes. Consequently, a cell-centered scheme is inherently five to six times more costly in terms of CPU time and memory than a cell-vertex scheme. However, for a given mesh, the cell-centered scheme is expected to produce a more accurate solution spatially since it is effectively solving the problem with five or six times the number of control volumes as the cell-vertex scheme. In either case, the spatial discretization involves a flux balance where the fluxes are computed along each face that makes up the control volume using flow variables that have been averaged. The averaged flow variables are determined using values at the three nodes which make up a given face for the cell-vertex approach and using values at the centroids of the two tetrahedra that share a given face for the cell-centered approach.

The unsteady Euler equations are a set of nondissipative hyperbolic conservation laws whose numerical solution requires some form of artificial dissipation to prevent oscillations near shock waves and to damp high frequency uncoupled error modes. Generally, a combination of harmonic and biharmonic operators is used, corresponding to second and fourth differences of the conserved variables, respectively. The simplest way to compute a harmonic operator in the cell-vertex approach, for example, is to sum the difference in flow variables over all edges which have a common vertex at node  $i$ , given by

$$\nabla^2 Q_i \approx \sum_{m=1}^M (Q_m - Q_i) \quad (2)$$

The biharmonic operator can be approximated similarly by applying Eq. (2) twice. This simple treatment of the artificial dissipation works reasonably well for meshes that are "regular" (smooth meshes with little distortion, and a similar number of edges share each node at nearly equal angles, etc.) For meshes that are not regular or when spatial adaption is used (naturally leading to highly irregular meshes), Eq. (2) does not work well. In these cases a more accurate treatment of the dissipative operators is required such as that given by

$$\nabla^2 Q_i \approx \sum_{m=1}^M w_m (Q_m - Q_i) \quad (3)$$

where the weights  $w_m$  have been derived by a constrained optimization so that Eq. (3) is exact for a linearly varying  $Q$  (thus implying second-order spatial accuracy) [20].

Advantages of the central-difference type discretization are that it is easier to code and takes less memory than an upwind discretization. However, the difficulties in calculating accurately the artificial dissipation on irregular unstructured grids, suggest that an upwind discretization may be more desirable. Also, the cell-centered approach has advantages in comparison with the cell-vertex approach. For example, the cell-centered scheme vectorizes naturally (no special coding is required) whereas the flux balance in a cell-vertex scheme involves vector recurrences which inhibit vectorization. To avoid vector recurrences and allow vectorization, cell-vertex schemes require that the cells be sorted into groups, such that not more than one cell in a group accesses a given node. The additional coding that is required to accomplish this is somewhat unwieldy and increases the computational cost of the cell-vertex approach. Consequently, the cost of a cell-vertex method is actually closer to the cost of a cell-centered method than the factor of five to six difference in control volumes would suggest.

### Upwind Algorithms

Upwind type algorithms have also been implemented as either cell-vertex or cell-centered discretizations. In the cell-vertex approach, the flow variables are stored at the nodes and the control volume is typically taken to be part of the neighboring cells which have a vertex at that node. In two dimensions, the part of the cells taken is determined by connecting the centroid of the cell with the midpoints of the two edges which share the node. In three dimensions, the part of the cells taken is determined by a surface that is constructed similarly. However, since this is somewhat complicated

geometrically to do in three dimensions, and since these surfaces (and their corresponding unit normals and areas) have to be recomputed at each time step when the mesh deforms for unsteady cases, the cell-vertex approach is not the preferred method. In the cell-centered approach, the flow variables are stored at the centroids of the cells and the control volume is simply the cell itself.

Cell-centered, upwind, spatial discretizations have been developed based on either the flux-vector splitting (FVS) of van Leer [21] or the flux-difference splitting (FDS) of Roe [22]. The FDS approach is generally preferable since it is less dissipative than FVS and is easier to code. Similar to the central-difference type algorithms, the upwind algorithms involve a flux balance where the fluxes along the four faces of a given tetrahedron are summed as

$$\sum_{m=1}^4 H \Delta S = \sum_{m=1}^4 (En_x + Fn_y + Gn_z) \Delta S \quad (4)$$

where  $\Delta S$  is the area of the face. With FDS the flux vector  $H$  is approximated by

$$H = \frac{1}{2} \left[ H(Q^+) + H(Q^-) - |\tilde{A}|(Q^+ - Q^-) \right] \quad (5)$$

where  $Q^-$  and  $Q^+$  are the state variables to the left and right of the cell face and  $A$  is the flux jacobian matrix given by  $\partial H / \partial Q$ . Also the tilde and the absolute value sign indicate that the flux jacobian is evaluated using the so-called Roe-averaged flow variables and the absolute value of the characteristic speeds. The left and right states  $Q^-$  and  $Q^+$ , are determined by upwind-biased interpolations of the primitive variables  $q$ , similar to that which is done on structured meshes.

A significant advantage of the upwind type discretization is that it is naturally dissipative, in comparison with central-difference discretizations, and consequently does not require adjustment of free parameters to control the dissipation. The upwinding accounts for the local wave-propagation characteristics of the flow and captures shock waves sharply with usually only one grid point within the shock structure.

## TEMPORAL DISCRETIZATIONS

The Euler equations have been integrated in time numerically using several different temporal discretizations. These discretizations are either of the explicit type along with several techniques to accelerate convergence to steady state or of the implicit type which generally do not require such additional techniques for rapid convergence to steady state.

### Explicit Algorithm

The explicit algorithm that has been developed for use on unstructured meshes is the standard, multi-stage, Runge-Kutta time integration scheme. In this scheme the flux balance is determined at each stage and, for computational efficiency, the artificial dissipation is evaluated only at the first stage. The scheme is second-order-accurate in time and includes the necessary terms to account for changes in cell volumes due to a moving or deforming mesh. Furthermore, this explicit-scheme has a step size that is limited by the Courant-Friedricks-Lewy (CFL) condition. To accelerate convergence to steady-state, the CFL number may be increased by averaging implicitly the residual with values at neighboring grid points. These implicit equations are solved approximately using several Jacobi iterations. Convergence to steady-state is further accelerated by local time stepping, and when using a central-difference type spatial discretization, by enthalpy damping. Enthalpy damping is normally not used with upwind spatial discretizations since these discretizations generally do not preserve enthalpy.

Advantages of the explicit type time integration are that it is simple to code and takes far less memory than an implicit time integration. Furthermore, when a central-difference spatial discretization is also employed, the convergence to steady state is fairly rapid and solutions are obtained with minimal computational resources. However, when finer meshes are used, either globally fine or locally fine due to spatial adaption, the rate of convergence slows dramatically. When an upwind spatial discretization is employed, the rate of convergence is also poor. What is required typically for rapid convergence on fine meshes is a multigrid strategy. Alternatively, an implicit temporal discretization may be used, especially since an implicit time integration is generally desirable for unsteady applications.

### Implicit Algorithm

The implicit algorithm that has been developed for use on unstructured meshes is based on the cell-centered, upwind, spatial discretization. The algorithm is derived in general by first linearizing the flux vector  $H$  according to

$$H^{n+1} = H^n + \frac{\partial H}{\partial Q} \Delta Q \quad (6)$$

where  $\partial H / \partial Q$  is the flux jacobian  $A$ , as discussed before, and  $\Delta Q = Q^{n+1} - Q^n$ . Linearizing both flux terms on the right-hand-side of Eq. (5) using Eq. (6), and ignoring the tilde on the flux jacobian, results in

$$\begin{aligned} & \left[ \frac{\text{vol}}{\Delta t} I + \sum_{m=1}^4 A^+(Q_j) \Delta S \right] \Delta Q_j + \sum_{m=1}^4 A^-(Q_m) \Delta S \Delta Q_m \\ & = -\frac{1}{2} \sum_{m=1}^4 \left[ H(Q^+) + H(Q^-) - |\tilde{A}|(Q^+ - Q^-) \right]^n \Delta S \end{aligned} \quad (7)$$

where  $I$  is the identity matrix, "vol" is the volume of the tetrahedron  $j$ , and  $\Delta Q_m$  is the change in flow variables in each of the four tetrahedra adjacent to tetrahedron  $j$ . Also in Eq. (7)  $A^+$  and  $A^-$  are forward and backward flux jacobians, respectively. For flux-difference splitting, the exact jacobian  $A$  (derivative of the right-hand side of Eq. (6) with respect to  $Q$ ) is too expensive to compute and thus an approximate jacobian is normally used. This is accomplished by constructing the jacobians making use of the fact that the forward and backward jacobians should have non-negative and non-positive eigenvalues (characteristic speeds), respectively.

Direct solution of the system of simultaneous equations which results from application of Eq. (7) for all tetrahedra in the mesh, requires the inversion of a large matrix with large bandwidth which is computationally expensive. Instead, a Gauss-Seidel relaxation approach is used to solve the equations whereby the summation involving  $\Delta Q_m$  is moved to the right hand side of Eq. (7). The terms in this summation are then evaluated for a given time step using the most recently computed values for  $\Delta Q_m$ . The solution procedure then involves only the inversion of a  $5 \times 5$  matrix (represented by the terms in square brackets on the left hand side of Eq. (7)) for each tetrahedron in the mesh. The method is implemented by re-ordering the list of tetrahedra that make up the unstructured mesh from upstream to downstream, and the solution is obtained by sweeping two times through the mesh as dictated by stability considerations. The first sweep is performed in the direction from upstream to downstream and the second sweep is from downstream to upstream.

Advantages of the implicit type time integration are that it is numerically stable for very large CFL numbers, even on very fine meshes, and consequently it enables rapid convergence to steady state. For unsteady applications, it allows the selection of the step size based on the physical problem rather than on numerical stability considerations. This is in contrast with an explicit time integration which has a restrictive step size for unsteady applications which is more severe on finer meshes. A disadvantage of the implicit Gauss-Seidel relaxation procedure is that it requires twice the memory of the explicit Runge-Kutta integration, primarily due to having to store the backward flux jacobian  $A^-$ . Alternative implicit temporal discretizations are being investigated currently to circumvent this problem.

## BOUNDARY CONDITIONS

To impose the flow tangency boundary conditions along the surface of the vehicle in the cell-centered, upwind spatial discretization method, the flow variables are set within dummy cells that are effectively inside the geometry being considered. The velocity components within the dummy cell are determined from the values in the cell adjacent to the surface. This is accomplished by first rotating the components into a coordinate system that has a coordinate direction normal to the boundary face. The sign of the velocity component in this direction is changed (hence imposing no flow through the face) and the three velocity components are then rotated back into the original  $x, y, z$  coordinate system. Also, pressure and density within the dummy cell are set equal to the values in the cell adjacent to the surface.

After application of the upwind-biased interpolation formula to determine  $q^-$  and  $q^+$  at each face, the velocity components are corrected to give a "strong" implementation of the surface boundary condition according to

$$\begin{aligned} u_{corrected} &= u - n_x(un_x + vn_y + wn_z) \\ v_{corrected} &= v - n_y(un_x + vn_y + wn_z) \\ w_{corrected} &= w - n_z(un_x + vn_y + wn_z) \end{aligned} \quad (8)$$

In the far field a characteristic analysis based on Riemann invariants is used to determine the values of the flow variables on the outer boundary of the grid. This analysis correctly accounts for wave propagation in the far field which is important for rapid convergence to steady-state and serves as a "nonreflecting" boundary condition for unsteady applications.

## RESULTS AND DISCUSSION

To demonstrate the speed and robustness of the cell-centered, upwind, implicit solution algorithm, calculations were performed for the Boeing 747 aircraft. The results were obtained using the unstructured mesh shown in Fig. 1. The 747

geometry includes the fuselage, the wing, horizontal and vertical tails, underwing pylons, and flow-through engine nacelles. The mesh for the 747 contains 101,475 tetrahedra and 19,055 nodes for the half-span airplane. Also there are 4,159 nodes and 8,330 triangles on the boundaries of the mesh which include the airplane, the symmetry plane, and the far field. Steady-state calculations were performed for the aircraft at a freestream Mach number of  $M_\infty = 0.84$  and an angle of attack of  $\alpha = 2.73^\circ$ . The results were obtained using a CFL number of infinity and the flux jacobians were updated only every twenty iterations.

The solution required approximately 3,500 CPU seconds (522 iterations) on the Cray-2 computer (Navier) at the National Aerodynamic Simulation Facility, NASA Ames Research Center, to converge the solution to plotting accuracy (taken to be a three order-of-magnitude reduction in the  $L_2$ -norm of the density residual). The resulting steady pressure coefficient contours on the surface of the 747 aircraft are shown in Fig. 2. The contours indicate that there is a significant amount of flow compression on the nose of the aircraft, along the inboard leading edge of the wing, and inside the cowl of the engine nacelles. There is flow expansion on the forward fuselage, on the horizontal and vertical tail surfaces, and on the upper surface of the wing terminated by a shockwave.

## CONCLUDING REMARKS

The paper summarized the development by the author of unstructured grid methods for the solution of the equations of fluid flow and shared some of the "lessons learned" over the course of the research. The focus of the discussion was on the solution of the three-dimensional Euler equations including central-difference and upwind-type spatial discretizations, explicit and implicit temporal discretizations, and strong and weak implementations of boundary conditions. A best choice of methods, in the opinion of the author, is a cell-centered, upwind, implicit solution algorithm. An example calculation with this algorithm, using a CFL number of infinity, was presented for the Boeing 747 aircraft. The results were obtained in less than one hour CPU time on a Cray-2 computer, thus demonstrating the speed and robustness of the present capability.

## ACKNOWLEDGEMENT

The author acknowledges Neal Frink of the Transonic Aerodynamics Branch, NASA Langley Research Center, Hampton, Virginia, and Paresh Parikh and Shahyar Pirzadeh of ViGYAN Research Associates, Hampton, Virginia, for providing the unstructured tetrahedral mesh for the Boeing 747 aircraft.

## REFERENCES

1. A. Jameson, AIAA Paper No. 87-1184, January 1987.
2. J. W. Edwards, and J. L. Thomas, AIAA Paper No. 87-0107, January 1987.
3. A. Jameson, and D. J. Mavriplis, *AIAA Journal*, vol. 24, April 1986, pp. 611-618.
4. D. J. Mavriplis, *AIAA Journal*, vol. 26, July 1988, pp. 824-831.
5. A. Jameson, T. J. Baker, and N. P. Weatherill, AIAA Paper No. 86-0103, January 1986.
6. K. Morgan, and J. Peraire, Von Karman Institute for Fluid Dynamics Lecture Series 1987-04, Computational Fluid Dynamics, March 2-6, 1987.
7. R. Lohner, *International Journal of Numerical Methods in Engineering*, vol. 24, 1987, pp. 1741-1756.
8. K. Morgan, J. Peraire, R. R. Thareja, and J. R. Stewart, AIAA Paper No. 87-1172, 1987.
9. J. Peraire, J. Peiro, L. Formaggia, and K. Morgan, AIAA Paper No. 88-0032, January 1988.
10. J. T. Batina, AIAA Paper No. 89-0115, January 1989.
11. R. D. Rausch, J. T. Batina, and H. T. Y. Yang, AIAA Paper No. 89-1384, April 1989.
12. J. T. Batina, AIAA Paper No. 89-1189, April 1989.
13. J. T. Batina, AIAA Paper No. 89-1816, June 1989.
14. E. M. Lee and J. T. Batina, NASA TM 102609, March 1990.
15. J. T. Batina, AIAA Paper No. 90-0936, April 1990.
16. J. T. Batina, Accuracy of Unstructured Grid Techniques Workshop, NASA Langley Research Center, Hampton, Virginia, January 16-17, 1990.
17. J. T. Batina, AIAA Paper No. 90-1649, June 1990.
18. W. L. Kleb, J. T. Batina, and M. H. Williams, AIAA Paper No. 90-1650, June 1990.
19. R. D. Rausch, J. T. Batina, and H. T. Y. Yang, AIAA Paper No. 91-1106, April 1991.
20. D. G. Holmes, and S. D. Connell, AIAA Paper No. 89-1392, June 1989.
21. B. Van Leer, *Lecture Notes in Physics*, vol. 170, 1982, pp. 507-512.
22. P. L. Roe, *Journal of Computational Physics*, vol. 43, 1981, pp. 357-372.

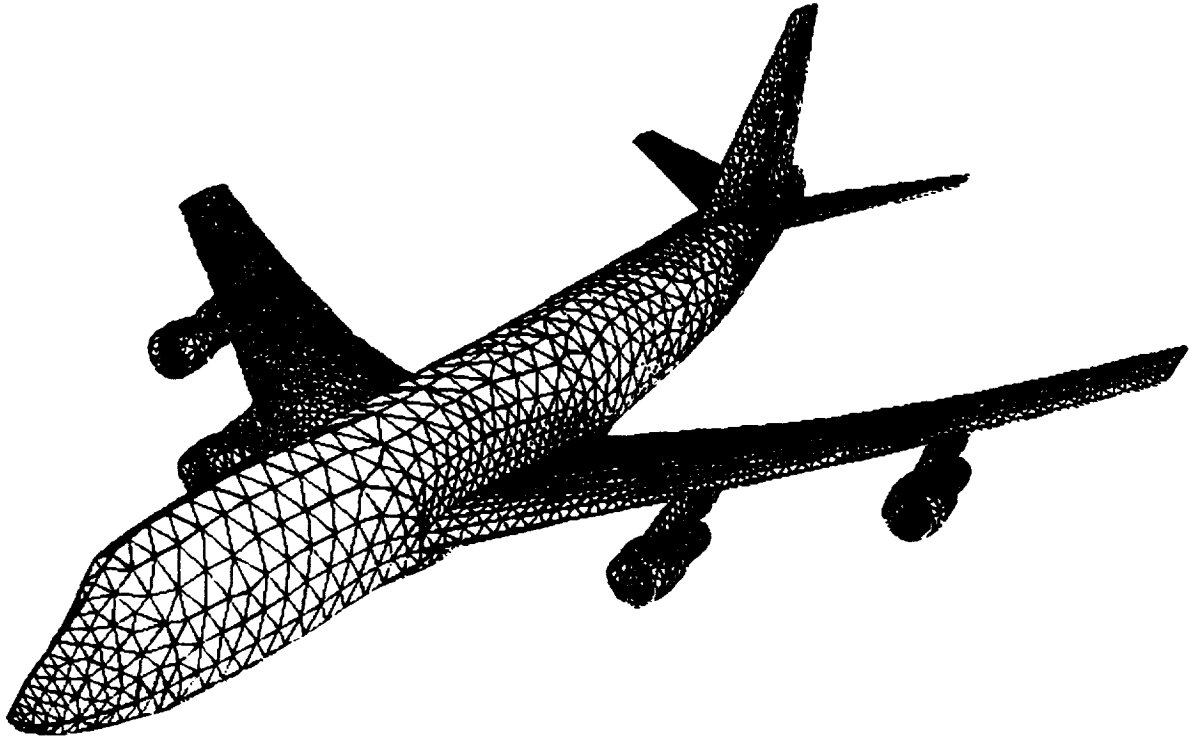


Fig. 1 Surface mesh of triangles for the Boeing 747 aircraft.

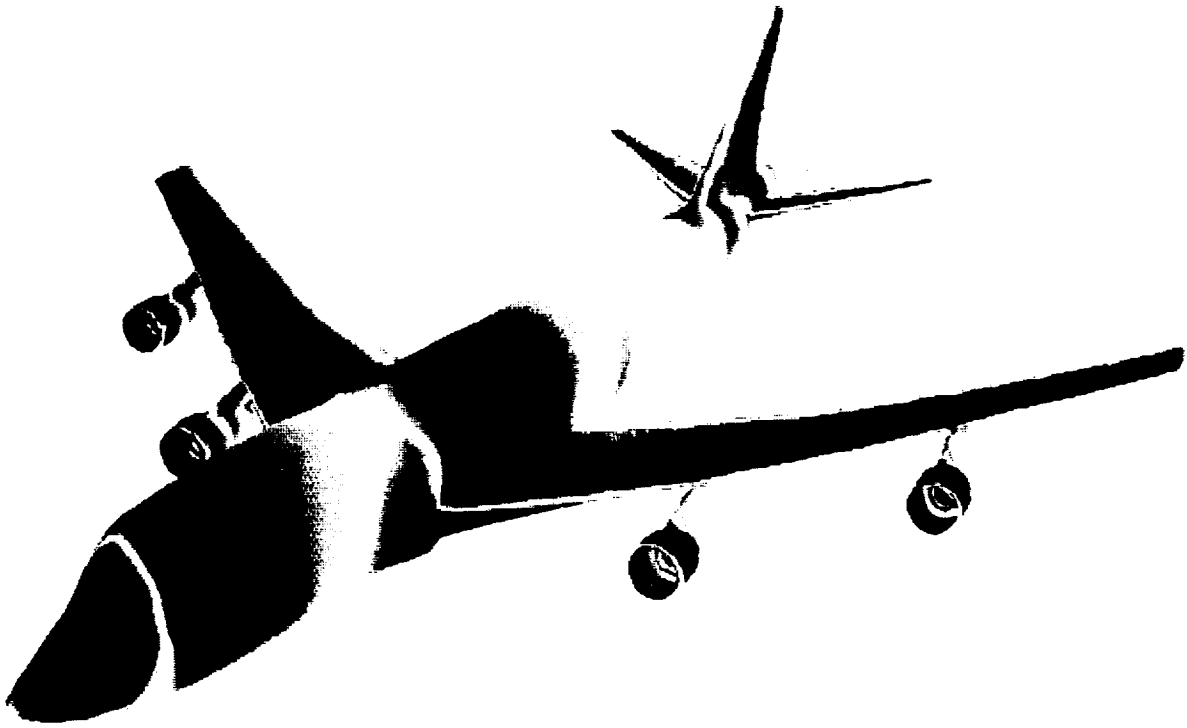


Fig. 2 Steady pressure coefficient contours on the Boeing 747 aircraft at  $M_\infty = 0.84$  and  $\alpha = 2.73^\circ$ .



# Report Documentation Page

|  |  |  |  |   |  |
|--|--|--|--|---|--|
| 1. Report No.<br><br>NASA TM-104137  |  | 2. Government Accession No.                              |  | 3. Recipient's Catalog No.  |  |
| 4. Title and Subtitle<br><br>Unstructured-Grid Methods Development - Lessons Learned   |  |  |  | 5. Report Date<br><br>August 1991   |  |
|  |  |  |  | 6. Performing Organization Code   |  |
| 7. Author(s)<br><br>John T. Batina   |  |  |  | 8. Performing Organization Report No.   |  |
| 9. Performing Organization Name and Address<br><br>NASA Langley Research Center<br>Hampton, Virginia 23665-5225  |  |  |  | 10. Work Unit No.<br><br>509-10-02-03   |  |
|  |  |  |  | 11. Contract or Grant No.   |  |
| 12. Sponsoring Agency Name and Address<br><br>National Aeronautics and Space Administration<br>Washington, DC 20546-0001   |  |  |  | 13. Type of Report and Period Covered<br><br>Technical Memorandum                     |  |
|  |  |  |  | 14. Sponsoring Agency Code  |  |
| 15. Supplementary Notes<br><br>Presented at the 4th International Symposium on Computational Fluid Dynamics, Davis, California, September 9-12, 1991.  |  |  |  |   |  |
| 16. Abstract<br><br>The paper summarizes the development by the author of unstructured grid methods for the solution of the equations of fluid flow and shares some of the "lessons learned" over the course of the research. The focus of the discussion is on the solution of the three-dimensional Euler equations including spatial discretizations, temporal discretizations, and boundary conditions. An example calculation with an upwind implicit method using a CFL number of infinity is presented for the Boeing 747 aircraft. The results were obtained in less than one hour CPU time on a Cray-2 computer, thus demonstrating the speed and robustness of the present capability. |  |  |  |   |  |
| 17. Key Words (Suggested by Author(s))<br><br>Computational Fluid Dynamics<br>Aerodynamics   |  |  |  | 18. Distribution Statement<br><br>Unclassified - Unlimited<br><br>Subject Category 02 |  |
| 19. Security Classif. (of this report)<br><br>Unclassified   |  | 20. Security Classif. (of this page)<br><br>Unclassified |  | 21. No. of pages<br><br>7   |  |
|  |  |  |  | 22. Price<br><br>A02  |  |

